EXPERIMENT - 24

Object Oriented Programming Lab

Aim

Write a program to overload unary increment (++) operator.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 24

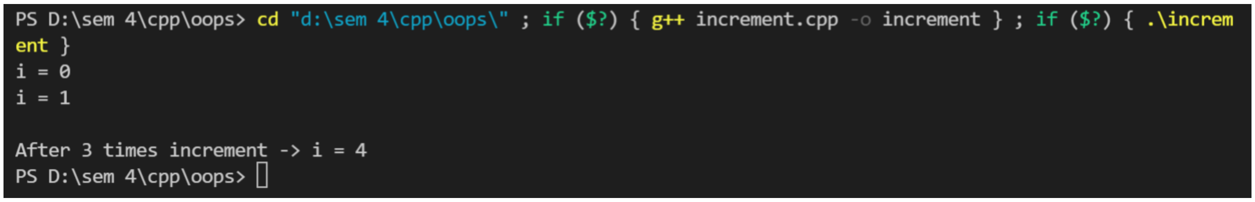
Aim:

Write a program to overload unary increment (++) operator.

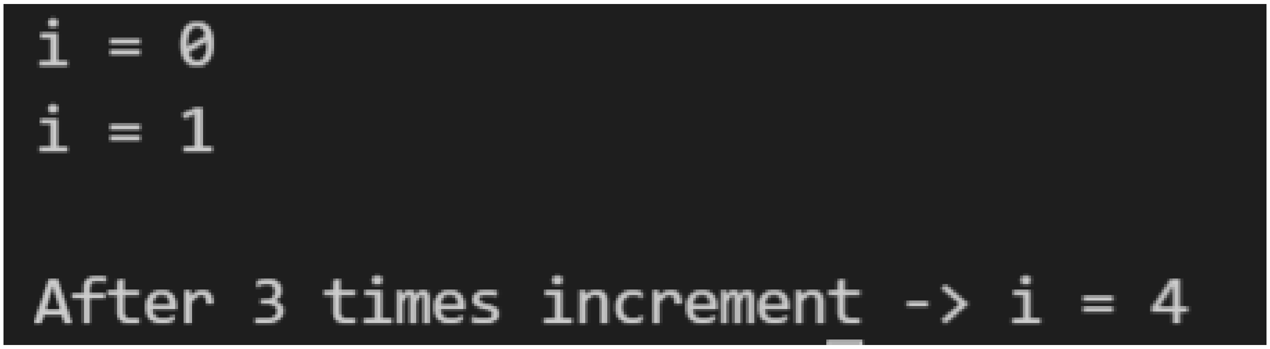
Source Code:

```
#include <iostream>
using namespace std;

class Check
{
private:
    int i;

public:
    Check() : i(0) {}
    void operator++()
    {
        ++i;
    }
    void Display()
    {
        cout << "i = " << i << endl;
    }
};

int main()
{
    Check obj;

    // Displays the value of data member i for object obj
    obj.Display();

    // Invokes operator function void operator ++( )
    ++obj;

    // Displays the value of data member i for object obj
    obj.Display();

    ++obj;
    ++obj;
    ++obj;
    cout << "\nAfter 3 times increment -> ";
    obj.Display();

    return 0;
}
```

Output:



```
PS D:\sem 4\cpp\oops> cd "d:\sem 4\cpp\oops\" ; if ($?) { g++ increment.cpp -o increment } ; if ($?) { .\increm
ent }
i = 0
i = 1

After 3 times increment -> i = 4
PS D:\sem 4\cpp\oops> 
```



```
i = 0
i = 1

After 3 times increment -> i = 4
```

Viva Questions

Q1) What are causes of function overloading?

Ans.

When the compiler is unable to decide which function is to be invoked among the overloaded function, this situation is known as **function overloading**.

When the compiler shows the ambiguity error, the compiler does not run the program.

Causes of Function Overloading:

- Type Conversion.
- Function with default arguments.
- Function with pass by reference.

Q2) What is overloading?

Ans.

C++ allows you to specify more than one definition for a **function** name or an **operator** in the same scope, which is called **function overloading** and **operator overloading** respectively.

An overloaded declaration is a declaration that is declared with the same name as a previously declared declaration in the same scope, except that both declarations have different arguments and obviously different definition (implementation).

When you call an overloaded ***function*** *or* ***operator****, the compiler determines the most appropriate definition to use, by comparing the argument types you have used to call the function or operator with the parameter types specified in the definitions. The process of selecting the most appropriate overloaded function or operator is called* ***overload resolution****.*

Q3) What is purpose of operator overloading?

Ans.

The purpose of operator overloading is to provide a special meaning of an operator for a user-defined data type. With the help of operator overloading, you can redefine the majority of the C++ operators. You can also use operator overloading to perform different operations using one operator.